Shlok S Iyer 1BM22CS260 sem:3 sec:CE

STACK IMPLIMENTATION

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node *next;
};
struct node *head;
void push(int new_data)

{

    struct node* new_node = (struct node*) malloc(sizeof(struct node));

    new_node->data = new_data;

    new_node->next = head;

    head = new_node;


}

void pop()

{

    struct node *ptr;

    if(head == NULL)

    {

        printf("\nList is empty ");

    }

    else

    {
```

```
        ptr = head;

        head = ptr->next;

        free(ptr);

        printf("\n Node deleted from the begining");

    }

}
void display()
{
    struct node *ptr;
    ptr=head;
    if(ptr==NULL)
    {
        printf("Nothing to print");
    }
    else
    {
        printf("\n printing the elements ...\n");
        while(ptr!=NULL)
        {
            printf("\n%d",ptr->data);
            ptr=ptr->next;
        }
    }

}
int main()
{
    int ch,val;
    struct node *head=NULL;

    while(1)
    {   printf("\n Enter the choice , 1. to push 2. pop, 3. ");
        scanf("%d",&ch);
        switch(ch)
    {
        case 1:
          {
```

```
            printf("Enter the value to input : ");
            scanf("%d",&val);
            push(val);
            break;
           }

        case 2:
           {
              pop();
              break;
           }
        case 3:
           {
              display();
              break;
           }
        case 4:
           {
              exit(1);
           }
        default:
           {
              printf("Invalid choice");
           }

    }
    }


}
```

```
 Enter the choice , 1. to push 2. pop, 3. 1
Enter the value to input : 1

 Enter the choice , 1. to push 2. pop, 3. 1
Enter the value to input : 2

 Enter the choice , 1. to push 2. pop, 3. 1
Enter the value to input : 3

 Enter the choice , 1. to push 2. pop, 3. 3

 printing the elements ...

3
2
1
 Enter the choice , 1. to push 2. pop, 3. 2

 Node deleted from the begining
 Enter the choice , 1. to push 2. pop, 3. 3

 printing the elements ...

2
1
 Enter the choice , 1. to push 2. pop, 3. 2

 Node deleted from the begining
 Enter the choice , 1. to push 2. pop, 3. 3
```

```
1
 Enter the choice , 1. to push 2. pop, 3. 2

 Node deleted from the begining
 Enter the choice , 1. to push 2. pop, 3. 2

List is empty
 Enter the choice , 1. to push 2. pop, 3.
```

QUEUE IMPLIMENTATION

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

struct node *front = NULL, *rear = NULL;

void enqueue(int val)
{
    struct node *newNode = malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = NULL;

    //if it is the first node
    if(front == NULL && rear == NULL)
        //make both front and rear points to the new node
        front = rear = newNode;
    else
    {
        //add newnode in rear->next
        rear->next = newNode;

        //make the new node as the rear node
        rear = newNode;
    }
}

void dequeue()
{
    //used to free the first node after dequeue
    struct node *temp;

    if(front == NULL)
        printf("Queue is Empty. Unable to perform dequeue\n");
    else
```

```
    {
        //take backup
        temp = front;

        //make the front node points to the next node
        //logically removing the front element
        front = front->next;

        //if front == NULL, set rear = NULL
        if(front == NULL)
            rear = NULL;

       //free the first node
       free(temp);
    }

}

void printList()
{
    struct node *temp = front;

    while(temp)
    {
        printf("%d->",temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main()
{
   int data, ch;
   printf("Menu:\n 1. Enqueue\n 2. Dequeue\n 3. Display\n 4. Exit");
   printf("\nEnter choice: ");
   scanf("%d",&ch);
   while(ch!=4){
    switch(ch){
    case 1:
        printf("Enter data to be pushed: ");
        scanf("%d",&data);
        enqueue(data);
        break;
    case 2:
```

```
        dequeue();
        break;
    case 3:
        printList();
        break;
    case 4:
        exit(0);
   }
   printf("\nEnter choice: ");
   scanf("%d",&ch);
   }

    return 0;
}
```

```
Menu:
 1. Enqueue
 2. Dequeue
 3. Display
 4. Exit
Enter choice: 1
Enter data to be pushed: 20

Enter choice: 1
Enter data to be pushed: 21

Enter choice: 1
Enter data to be pushed: 22

Enter choice: 1
Enter data to be pushed: 23

Enter choice: 1
Enter data to be pushed: 24

Enter choice: 3
20->21->22->23->24->NULL

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 2
Queue is Empty. Unable to perform dequeue

Enter choice: 3
NULL
```